Transaction Collision Mitigation in DAG Blockchains: the Weak-Block Signaling Approach

Canhui Chen*† Zhixuan Fang*†
*Tsinghua University, Beijing, China
†Shanghai Qi Zhi Institute, Shanghai, China
chen-ch24@mails.tsinghua.edu.cn, zfang@mail.tsinghua.edu.cn

***Abstract*—DAG-based blockchains face the key challenge of transaction inclusion collision due to the high concurrency and network delay. In this paper, we propose "We-TIPS", a weak-block-based transaction inclusion protocol with signaling, designed to tackle this key challenge. In We-TIPS, during the mining process, the miner can broadcast the weak block header as a signal, which can indicate the miner's current transaction inclusion. With the prompt broadcast of the signal, the miner can effectively avoid the transaction inclusion collision and thus greatly boost the system performance. Besides, we develop a transaction inclusion game in We-TIPS to model miners' interaction and show that it is a potential game. We propose a decentralized transaction inclusion algorithm that can achieve the approximate Nash equilibrium. Finally, we conduct intensive experiments to demonstrate the superior performance of We-TIPS.**

***Index Terms*—blockchain network, performance analysis, game theory**

I. Introduction

The Internet of Things (IoT) and wireless networks are rapidly transforming the way we interact with our environment, leading to a vast and interconnected network of devices, sensors, and services [2], [3]. With the increased connectivity, security, and reliability of these networks, there is a growing need for a distributed and decentralized approach to managing the vast amounts of data generated by IoT devices [4]. This is where blockchain technology comes into play [5]. Blockchain technology has emerged as a potential solution to address the challenges of IoT data management by providing a secure and decentralized platform for data exchange [6]–[8]. However, the traditional blockchain architecture requires the explicit confirmation of each block, which can result in delays and inefficiencies in wireless networks with low bandwidth and high latency. To overcome these limitations, a new type of blockchain architecture has emerged, called the Directed Acyclic Graph (DAG)-based blockchain [9]. The DAG-based blockchain ensures that multiple transactions can be processed simultaneously across multiple blocks without the need for global confirmation, which is important for IoT applications.

While DAG-based blockchains offer many advantages for wireless networks and IoT applications, one of the key challenges they face is transaction inclusion collision [10]. This occurs due to high concurrency and network delays, as miners may not have access to the most up-to-date information about the blockchain, especially in wireless networks with low bandwidth and high latency. As a result, the same transactions may be included in concurrent blocks, leading to redundant records in the blockchain. This collision in transaction inclusion can waste block capacity and severely degrade system performance, posing a significant challenge for the effective implementation of DAG-based blockchains in wireless networks and IoT applications [11].

Faced with this challenge, authors in [12] propose "TIPS", a transaction inclusion protocol with signaling in the DAG-based blockchain. TIPS [12] includes a Bloom filter in the block header which can indicate the included transactions, and broadcast the block header as a signal when a miner successfully mines a new block. Since the size of the signal is small enough, the signal can be broadcast to the whole blockchain network in a short time and effectively help the miners to avoid transaction inclusion and thus significantly improve the system performance. However, we notice that TIPS only signals other miners when a new block is successfully mined, which limits its performance improvement. To address this problem, we propose "We-TIPS", the weak-block-based transaction inclusion protocol with signaling in DAG-based blockchain, which can signal the miners during the mining process to avoid transaction inclusion collision.

Before successfully mining a new block (i.e. finding the solution to the hashing puzzle), the miner may find several weak solutions that do not satisfy the full mining difficulty requirement but also reflect a valid partial proof-of-work (PoW). The block with the weak solution is called "weak block". This is similar to the partial proof-of-work, i.e., "shares" in mining pools [13]. Similar to TIPS [12], we include the Bloom filter into the block header so that a block header can serve as a signal indicating the transactions included in the block. Differently, instead of broadcasting the signal only when a new block is mined, We-TIPS allows the miner to broadcast the weak block header as a signal during the mining process. With the prompt broadcast of the signal in the weak block header, the miner can know other miners' current transaction inclusion strategies, based on which, the miner can adjust his mining strategy to avoid transaction inclusion collision. Besides, to model the miners' interaction in We-TIPS, we develop a transaction inclusion game. We

Part of the results appeared in WiOpt 2023 [1]. This work is supported by Tsinghua University Dushi Program and Shanghai Qi Zhi Institute Innovation Program.

Corresponding author: Zhixuan Fang at Tsinghua University (Email: zfang@mail.tsinghua.edu.cn).

show that the transaction inclusion game in We-TIPS is a potential game. We further propose a decentralized transaction inclusion algorithm that can achieve the approximate Nash equilibrium. Both the theoretical analysis and experimental validation support the high efficiency of the We-TIPS.

The key contributions of the paper are as follows:

- We propose a novel weak-block-based transaction inclusion protocol with signaling (We-TIPS) in the DAG-based blockchain. We-TIPS allows miners to broadcast their weak block header as a signal during the mining process, which can indicate the miners' current transaction inclusion strategies and help the miners to avoid the transaction inclusion.
- We adopt a game-theoretic framework to model the miners' interaction in We-TIPS. We show that the transaction inclusion game in We-TIPS can be modeled as a potential game. Besides, we propose a decentralized transaction inclusion algorithm that can achieve the approximate Nash equilibrium.
- We conduct a rigorous theoretical analysis encompassing both sensitivity and security aspects. The sensitivity analysis provides a theoretical perspective on the influence of We-TIPS and the weak block ratio on system performance. Concurrently, the security analysis demonstrates both incentive compatibility and system robustness.
- We provide some empirical results of the existing DAG-based blockchain to demonstrate how transaction inclusion collision degrades the system performance in reality. Besides, we develop a DAG-based blockchain simulator and conduct intensive experiments. The experiment results show that We-TIPS can achieve the nearly-optimal performance. Specifically, We-TIPS can achieve more than 98% utilization, while "TIPS" can achieve 90% utilization and the current popular protocol "Conflux" only achieves 72% utilization.

The rest of the paper is organized as follows. In Section II, we introduce the background of the DAG-based blockchain. In Section III, we introduce the system model of We-TIPS. In Section IV, we develop the transaction inclusion game in We-TIPS, and propose a transaction inclusion algorithm to achieve the approximate Nash equilibrium. In Section V, we delve into the dynamics of transaction inclusion strategies among miners with heterogeneous hash rates. In Section VI, we conduct a sensitivity analysis on the weak block ratio to investigate how variations in this parameter affect system performance. In Section VII, we demonstrate the incentive compatibility and system robustness of We-TIPS. In Section VIII, we conduct intensive experiments to demonstrate the efficiency of We-TIPS. In Section IX, we discuss several practical aspects of We-TIPS. In Section X, we review related literature. Section XI concludes the paper with the final remark.

II. Background

A. Overview of DAG-Based Blockchains

A Directed Acyclic Graph (DAG) is a graph structure that consists of vertices (or nodes) connected by edges with a direction where no cycles exist. DAG-based blockchains use this structure to record transactions more flexibly than traditional linear blockchains [9].

In a conventional linear blockchain (such as Bitcoin [14]), transactions are grouped into blocks that are arranged sequentially. Each block contains a block header and a block body. The block header includes metadata, such as a timestamp, a reference to the previous block's hash, and a Merkle root summarizing the transactions. The block body carries the actual transactions. The linearity of the chain requires blocks to be added sequentially, which can lead to transaction processing bottlenecks, high latency, and limited throughput.

DAG-based blockchains [9] allow multiple blocks (or transactions) to be processed concurrently. In these systems, the ordering of transactions is determined by the DAG structure rather than a single linear sequence. This concurrency can lead to improvements in scalability and efficiency, as more transactions can be processed in parallel.

B. Key Components in DAG-based Blockchains

a) Block Header: The block header is a concise summary of the block's contents. It typically contains a reference (or hash) to the previous block (or blocks, in the case of DAG systems), a timestamp, and other metadata (such as the nonce in proof-of-work systems). The smaller size of the header compared to the full block body enables faster network propagation.

b) Blockchain Mining: Mining is the process by which new blocks are produced and added to the blockchain. In proof-of-work (PoW) systems [14], miners compete by repeatedly hashing the block header with varying nonce values until they find a hash that meets a predetermined difficulty target. The miner who finds a valid solution first broadcasts the block, and other network participants verify its correctness before accepting it. Importantly, each block header includes a Proof of Work that is inherently linked to the miner's address, meaning that the mining puzzle is unique to each participant. This guarantees that each miner's work remains both secure and distinct. Mining thus serves two key purposes: it secures the network by making block creation computationally expensive and it determines the block production rate. In DAG-based blockchains that inherit the PoW mechanism (e.g., block–DAG systems) [9], the mining process is similar to that in linear blockchains, but the structure of the ledger allows multiple blocks to be produced concurrently.

c) Transaction Inclusion and Collision: Transaction inclusion refers to the process whereby miners select transactions from a pool and incorporate them into blocks. In both linear and DAG-based blockchains, multiple miners may choose to include the same transaction, leading to what is known as a transaction inclusion collision [10]. When collisions occur, the same transaction may be redundantly included by several miners, thereby reducing the effective throughput of the system, defined as the number of distinct transactions confirmed in a given time. Efficient mechanisms to coordinate transaction selection and mitigate collisions are especially critical in high-concurrency environments.

C. Two Types of DAG-Based Blockchains

DAG-based blockchain systems can be broadly divided into two categories [9]:

a) Block-DAG Systems: In block-DAG architectures, blocks are organized in a Directed Acyclic Graph (DAG) instead of a traditional linear chain. Notable examples of block-DAG systems include Conflux [15], Spectre [16], and Phantom [17]. This arrangement allows for the concurrent generation of multiple blocks, thereby enhancing the overall throughput of the network. By aggregating numerous transactions into a single block, block-DAG systems efficiently compress data, which in turn improves system performance and reduces overhead associated with transaction verification.

b) Transaction-DAG Systems: Transaction-DAG systems, on the other hand, take a more granular approach by treating individual transactions as vertices within the DAG structure. Prominent systems implementing this paradigm include Byteball [18], Nano [19], and Avalanche [20]. This allows for greater flexibility in transaction handling and verification, as each transaction is propagated and validated independently. Despite their flexibility, the absence of block aggregation means transaction-DAG systems may suffer from increased network and validation overhead, which can negatively impact overall system efficiency.

In this paper, we concentrate exclusively on block-DAG systems that incorporate Bitcoin's Proof of Work (PoW) mining paradigm. Our research addresses the inefficiencies stemming from transaction inclusion collisions, a prevalent issue that occurs when miners, due to network latency and high concurrency, incorporate identical transactions into their respective blocks. Within a block-DAG framework, a miner's block header, being relatively lightweight, can function as a signal to other miners. This signal can be disseminated swiftly across the network, facilitating coordinated transaction inclusion and thereby reducing collisions. Conversely, in transaction-DAG systems, the overhead of broadcasting an additional signal is nearly equivalent to broadcasting the transaction itself. Consequently, the signaling mechanism proposed in our study would provide only marginal advantages in transaction-DAG scenarios. Therefore, we limit our analysis and protocol design to block-DAG systems, where the benefits of rapid and compact signals are most substantial.

III. System Model

In this section, we introduce the system model of "We-TIPS", i.e., the weak-block-based transaction inclusion protocol with signaling in the DAG-based blockchain. The key feature of We-TIPS is that the miners in We-TIPS can broadcast the signal using a weak block header indicating their current transaction selection during the mining process, by which the miners can coordinate their transaction inclusion strategies to avoid transaction inclusion collision.

A. Model Overview

Figure 1 demonstrates the system model of We-TIPS. In We-TIPS, during the mining process, each miner may find some weak headers, which are the partial solutions to the mining puzzle. There is a Bloom filter [21] in the header, which can indicate the transactions that the miner is currently mining on. Therefore, the weak header can serve as a signal during the mining process. Once a miner finds a weak header, he will broadcast his weak header to other nodes. And when receiving weak headers from other nodes, the miner will adjust his transaction inclusion strategy to avoid transaction inclusion collision. Once a miner successfully solves the mining puzzle (finds a strong header), he will follow the TIPS protocol [12], that is, he will broadcast the strong header first to further avoid transaction inclusion collision.

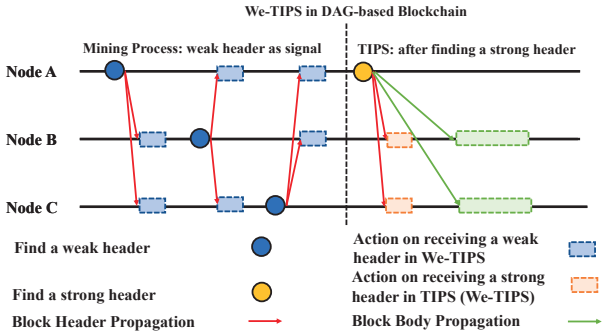


Fig. 1. System model of We-TIPS

B. Block Layout

In the PoW-based blockchain system, every miner tries to solve a PoW puzzle by computing the hash function over a newly created header. The header is constantly being changed by modifying its nonce field, until a valid hash value is found. Consistent with the definitions of [22], we denote the block header that solves the PoW puzzle as a "strong header" with its hash value h smaller than the strong target (mining difficulty) T_s. The "weak header" is a block header whose hash value does not meet the strong target T_s, but is still low enough to prove a significant PoW, i.e. $T_s \leq h < T_w$, where $T_w > T_s$ and T_w is the weak target. For convenience, we denote $\beta = T_w/T_s \geq 1$ as the weak block ratio in We-TIPS, which demonstrates the expected number of weak/strong block headers corresponding to each strong block header. Specifically, $\beta = 1$ indicates the scenario without weak blocks, where the proposed We-TIPS reduces to TIPS [12]. Similar to TIPS [12], we insert the Bloom filter in the block header. A Bloom filter is a space-efficient probabilistic data structure, conceived by Burton Howard Bloom in 1970 [21], which is used to test whether an element is a member of a set. We can consider the block header with the Bloom filter as a signal in our system, where the Bloom filter can indicate the selected transactions in the block. As discussed in TIPS [12], the size of the Bloom filter is small enough so that the block header can be propagated through the whole network in a short time.

C. Mining Process

The mining process is shown in Algorithm 1. During the mining process in We-TIPS, the miners' behaviors of mining and receiving the strong block header and block body are

Algorithm 1: Mining process in We-TIPS

```
on MineBlock:
    for nonce ∈ {0, 1, 2, ···} do
        header ← createHeader(nonce)
        hash ← H(header)
        if hash < T_s then
            // strong header
            broadcast(header)
            broadcast(body)
            return
        if hash < T_w then
            // weak header
            broadcast(header)
on ReceiveBlockHeader⟨header⟩
    hash ← H(header)
    assert(validHeader(header) and hash < T_w)
    BF ← getBloomFilter(header)
    // select the transactions hitting BF from transaction pool
    TX ← getTransactionSet(BF, txpool)
    if hash < T_s then
        // strong header
        Update transaction pool based on TIPS [12]
    if hash < T_w then
        // weak header
        Update the selecting transactions based on Algorithm 2
on ReceiveBlockBody⟨body⟩
    header ← getBlockHeader(body)
    hash ← H(header)
    // we only handle the block body with a strong header
    assert(validBody(body) and hash < T_s)
    Update transaction pool based on TIPS [12]
```

consistent with those in TIPS. And the miner's behaviors on weak block headers are summarized as follows:

- When a miner finds a weak block header, it only needs to broadcast the weak block header.
- When a miner receives a weak block header, it needs to check the validation of the weak block header. If the weak block header is valid, the miner will update his transaction inclusion strategy to avoid transaction inclusion collision, which will be discussed in Section IV.

Note that the weak block header does not change the consensus protocol in the DAG-based blockchain system. Thus We-TIPS can be used as an "add-on" component, and can be applied to most of the current DAG-based blockchain protocols.

IV. Transaction Inclusion Strategy

In this section, we investigate the miners' transaction inclusion strategies in We-TIPS. We first show that the miners' interactions can be modeled as a multi-agent Markov decision process (MMDP). To model the miner's rationality, we investigate the miner's transaction inclusion strategy from a game-theoretical perspective. We show that the transaction inclusion game in We-TIPS is a potential game and further propose a transaction inclusion algorithm that can achieve the approximate Nash equilibrium with a good system performance.

A. Blockchain Modeling

We consider a DAG-based blockchain system with N miners, where the block generation process follows the Poisson process with a rate λ. We denote the effective network propagation delay as Δ. Each miner maintains a transaction pool containing at most m transactions. Due to the block size limit, each block can contain at most n transactions. Without loss of generality, we assume that the transactions in the memory pool are sorted in descending order by their transaction fees, and the transaction fee of the transaction i is denoted as f_i. Each miner is rational and will choose his transaction inclusion strategy to maximize his utility, which forms a transaction inclusion game in We-TIPS.

B. Miners' Behaviors Modeling

Since the block generation process follows the Poisson process, which satisfies the memorylessness property, the miners' behavior can be modeled as a multi-agent Markov decision process (MMDP).

1) Multi-agent Markov Decision Process: The multi-agent Markov decision process (MMDP) of the mining process can be represented as a tuple $< \mathcal{N}, \mathcal{S}, (A_i)_{i\in\mathcal{N}}, P, (R_i)_{i\in\mathcal{N}} >$, where $\mathcal{N}$ denotes the set of N miners in the DAG-based blockchain system, with x_i denoting the hash rate of miner i and $\mathbf{x} = [x_1, \ldots, x_N]$ denoting the normalized hash rate vector, i.e., $\sum_{i=1}^{N} x_i = 1, x_i \in [0, 1]$; $\mathcal{S}$ is a set of states, each state $s \in \mathcal{S}$ can be represented as a tuple $< \mathbf{f}, W, \mathbf{b} >$, where $\mathbf{f} \in \mathbb{R}^{1\times m}$ denotes the transaction fees of m transactions in the memory pool, and $W \in \mathbb{R}^{N\times m}$ is a matrix denoting the miners' transaction selections in their previous weak headers. Specifically, $W(i, j) = 1$ denotes that the latest weak header mined by miner i contains transaction j [1], otherwise $W(i, j) = 0$. The symbol $\mathbf{b} \in \mathbb{R}^{N\times 1}$ denotes the mining results. $b_i = 1$ implies that the miner i successfully solves the PoW puzzle, i.e., finds the strong block header. Otherwise $b_i = 0$. The symbol $A_i \in \mathbb{R}^{1\times m}$ denotes the action of miner i. Specifically, $A_{i,j} = 1$ implies that the miner i includes the transaction j in a block and mines on that. Otherwise, we have $A_{i,j} = 0$. Since there are at most n transactions in a block, we have $||A_i||_1 = \left(\sum_{j=1}^{m} A_{i,j}\right) \leq n, \forall i \in \mathcal{N}$, where $||\cdot||_1$ is the Manhattan norm of a vector. Typically, we consider the case when $||A_i||_1 = n, \forall i \in \mathcal{N}$. Furthermore, the joint actions of miners can be represented as $A = A_1 \times \ldots \times A_N \in \mathbb{R}^{N\times m}$. $R_i(s, A)$ is the reward of miner i at state s with the joint actions A. P is the probability transition function that describes state transition, conditioned on the past states s and joint actions A. This game satisfies the Markov property due to the memorylessness of the mining process, i.e., $P[s_{t+1}|s_t, A_t, \ldots, s_0, a_0] = P[s_{t+1}|s_t, A_t]$.

2) Reward: Since the coinbase transaction reward is independent of the miner's transaction selection, for simplicity, we only consider the transaction fee reward in the miner's revenue. And the miner can obtain the transaction fee reward only when he successfully mines a strong block. Thus, the weak block header in We-TIPS only serves as a signal, and miners cannot get any reward from mining a weak block header. Besides, if there exist ι miners who have successfully mined a strong block with the same transaction i during the network propagation period Δ, we assume that the expected

[1]When we say the block header contains transaction j, it implies that the transaction j hits the Bloom Filter in the block header.

reward for any one of these miners on the transaction i is f_i/ι, i.e., assuming an equal network advantage for all miners. Note that such a model of probabilistic and homogeneous network advantage during the propagation period is common in the previous study (e.g., [10], [23]).

3) State Transition: We consider the state that at least one of the miners successfully solves the PoW puzzle and finds the strong block header as the final state, i.e., $s^* =< \mathbf{f}, W, \mathbf{b} >$ where $\mathbf{b} \neq \vec{\mathbf{0}}$ is a final state. The game ends at the final state without any further state transitions and will restart at the next step. We call the other states with $\mathbf{b} = \vec{\mathbf{0}}$ as processing states. With $\beta = T_w/T_s$ denoting the difficulty ratio of mining a strong block and weak block, the probability of transition from the processing states to the final states is $1/\beta$, and the probability of transition from the processing states to other processing states is $1-1/\beta$. Specifically, the probability of miner i mining a weak header, and the state transfer from the processing state $s =< \mathbf{f}, W, \vec{\mathbf{0}} >$ to the processing state $s^{(i)} =< \mathbf{f}, W^{(i)}, \vec{\mathbf{0}} >$ is $P\left(< \mathbf{f}, W^{(i)}, \vec{\mathbf{0}} > | < \mathbf{f}, W, \vec{\mathbf{0}} >, A\right) = \left(1 - \frac{1}{\beta}\right) x_i$, where $W^{(i)}(j,k) = \begin{cases} W(j,k), & j \neq i, \\ A(i,k), & j = i. \end{cases}$ The probability that at least one of the miners solves the PoW puzzle and mines a strong block, and the state transfer from the processing state $s =< \mathbf{f}, W, \vec{\mathbf{0}} >$ to the final state $s^* =< \mathbf{f}, W, \mathbf{b} >$ is formulated in the following lemma:

Lemma 1. *The probability that the state transfers from the processing state $s =< \boldsymbol{f}, W, \vec{\boldsymbol{0}} >$ to the final state $s^* =< \boldsymbol{f}, W, \boldsymbol{b} >$ is*

$$P\left(s^* | s, A\right) = \frac{1}{\beta} \left(\frac{(\lambda\Delta)^{||\boldsymbol{b}||_1 - 1}}{(||\boldsymbol{b}||_1 - 1)!} e^{-\lambda\Delta} \right) \sum_{\pi \in \mathcal{P}^{\mathcal{K}}} \prod_{i=1}^{|\mathcal{K}|} \frac{x_{\pi(i)}}{1 - \sum_{j=1}^{i-1} x_{\pi(j)}},$$

where $\mathcal{K} = \{i | b_i = 1\}$ denotes the set of miners who generate new blocks at the final state s^*.

Proof. At the final state $s^* =< \mathbf{f}, W, \mathbf{b} >$, there are $||\mathbf{b}||_1$ blocks generated in the blockchain system. Then $(||\mathbf{b}||_1 - 1)$ blocks are generated during the block propagation time Δ. Since the block generation process follows the Poisson process with a rate λ, the probability that $(||\mathbf{b}||_1 - 1)$ blocks are generated during the block propagation time Δ is $\frac{(\lambda\Delta)^{||\mathbf{b}||_1 - 1}}{(||\mathbf{b}||_1 - 1)!} e^{-\lambda\Delta}$. Denote the set $\mathcal{K} = \{i | b_i = 1\}$ as the set of miners who generates a new block at the final state s^*, and we have $|\mathcal{K}| = ||\mathbf{b}||_1$ and $\mathcal{K} \subset \mathcal{N}$. Then the problem of calculating the probability that the miners in $\mathcal{K}$ mine new blocks given the condition that $|\mathcal{K}|$ blocks will be generated is equivalent to the sampling problem with varying probability without replacement [24], where the normalized miners' hash rates $\mathbf{x}$ corresponds to their weights. Let $\mathcal{P}^{\mathcal{K}}$ denote the set of all permutations of the elements of $\mathcal{K}$. Then the probability that the miners in $\mathcal{K}$ mine new blocks given the condition that $|\mathcal{K}|$ blocks will be generated is

$$\Pr(\mathcal{K}) = \sum_{\pi \in \mathcal{P}^{\mathcal{K}}} \prod_{i=1}^{|\mathcal{K}|} \frac{x_{\pi(i)}}{1 - \sum_{j=1}^{i-1} x_{\pi(j)}}. \tag{1}$$

Since the probability of transition from the processing states to the final states is $1/\beta$, the probability that the state transfer from the processing state $s =< \mathbf{f}, W, \vec{\mathbf{0}} >$ to the final state $s^* =< \mathbf{f}, W, \mathbf{b} >$ is

$$P(s^* | s, A) = \frac{1}{\beta} \left(\frac{(\lambda\Delta)^{||\mathbf{b}||_1 - 1}}{(||\mathbf{b}||_1 - 1)!} e^{-\lambda\Delta} \right) \sum_{\pi \in \mathcal{P}^{\mathcal{K}}} \prod_{i=1}^{|\mathcal{K}|} \frac{x_{\pi(i)}}{1 - \sum_{j=1}^{i-1} x_{\pi(j)}}.$$

The proof is thus completed. □

C. Game Analysis

For simplicity, in the following discussion, we assume that the miners are homogeneous to enable tractable analysis. We further show that the transaction inclusion game in We-TIPS is a potential game.

When the miners are homogeneous, we know that the expected reward for a miner to include a certain transaction depends on the number of miners who also select the same transaction, which is formulated in the following lemma.

Lemma 2. *The expected reward for a miner to include transaction j given that there are total c miners who decide to include transaction j in their newly-mined block is*

$$r_j(c) = \sum_{k=0}^{\infty} \left((\lambda\Delta)^k e^{-\lambda\Delta} \left(\prod_{i=0}^{k-1} (N - 1 - i) \right)^{-1} \cdot \sum_{t=0}^{\min(c-1,k)} \binom{c-1}{t} \binom{N-c}{k-t} \frac{f_j}{t+1} \right).$$

Proof. Denote the set $\mathcal{K}$ as the set of miners (except itself) who generates a new block during the block propagation time Δ. Since the block generation process follows the Poisson process with a rate λ, the probability that k blocks are generated during the block propagation time Δ is

$$\Pr(|\mathcal{K}| = k) = \frac{(\lambda\Delta)^k}{k!} e^{-\lambda\Delta}.$$

Then the probability that the miners in $\mathcal{K}$ mine new blocks given the condition that $|\mathcal{K}|$ blocks will be generated can be calculated using equation (1). As miners are homogeneous, they have the same hash rate, i.e., $x_i = 1/N, \forall i \in \mathcal{N}$. Then equation (1) can be simplified as

$$\Pr(\mathcal{K} \mid |\mathcal{K}| = k) = k! \left(\prod_{i=0}^{k-1} (N - 1 - i) \right)^{-1}.$$

Denote the set $\mathcal{C}$ as the set of miners (except itself) who decide to include transaction j in their newly-mined block, then we have $|\mathcal{C}| = c - 1$ and $\mathcal{C} \subset \mathcal{N}$. Therefore, at the final state s^*, there will be $|\mathcal{K} \cap \mathcal{C}|$ miners who include transaction j in their newly-mined block. Then the expected reward for these miners on transaction j is $f_j / (|\mathcal{K} \cap \mathcal{C}| + 1)$. Let $t = |\mathcal{K} \cap \mathcal{C}|$, then the number of combinations that leads to the expected reward of $f_j / (t+1)$ is

$$\begin{aligned} \#\text{Comb}(t) &= \binom{|\mathcal{C}|}{|\mathcal{K} \cap \mathcal{C}|} \cdot \binom{|\mathcal{N} - \mathcal{C}| - 1}{|\mathcal{K}| - |\mathcal{K} \cap \mathcal{C}|} \\ &= \binom{c-1}{t} \cdot \binom{N-c}{k-t}. \end{aligned} \tag{2}$$

Therefore, the expected reward for a miner to include transaction j given that there are total k miners who decide to include transaction j in their newly-mined block is

$$\begin{aligned} r_j(c) &= \sum_{k=0}^{\infty}\left(\Pr(|\mathcal{K}|=k)\cdot\Pr(\mathcal{K})\cdot\sum_{t=0}^{\min(c-1,k)}\#\text{Comb}(t)\right) \\ &= \sum_{k=0}^{\infty}\left(\frac{(\lambda\Delta)^k}{k!}e^{-\lambda\Delta}\cdot k!\left(\prod_{i=1}^{k-1}(N-1-i)\right)^{-1}\right. \\ &\quad\left.\cdot\sum_{t=0}^{\min(c-1,k)}\binom{c-1}{t}\binom{N-c}{k-t}\frac{f_j}{t+1}\right). \end{aligned}$$

The proof is thus completed. □

With the expected reward analysis, we can further show that the transaction inclusion game in We-TIPS is a potential game.

Theorem 1. *The transaction inclusion game in We-TIPS is a potential game, where the utility for the miner i with joint action A is*

$$u_i(A) = \sum_{j\in\{k|A_{i,k}=1\}} r_j(c_j(A)),$$

where $c_j(A)$ denotes the number of miners including the transaction j given the joint action A, and $r_j(c)$ denotes the expected reward on transaction j given that there are totally c miners selecting the transaction j, which is formulated in Lemma 2, and the potential function is $\Phi(A) = \sum_{j=1}^{m}\sum_{k=1}^{c_j(A)} r_j(k)$, where m is the number of transactions in the transaction pool.

Proof. Consider the case where a single miner changes its strategy from A_i to B_i. Let Δu_i be the change in its cost caused by the change in strategy, then we have

$$\begin{aligned} \Delta u_i &= u_i(B_i, A_{-i}) - u_i(A_i, A_{-i}) \\ &= \sum_{j\in\{k|B_{i,k}=1\wedge A_{i,k}=0\}} r_j(c_j(A)+1) - \sum_{j\in\{k|B_{i,k}=0\wedge A_{i,k}=1\}} r_j(c_j(A)) \end{aligned}$$

Let $\Delta\Phi$ be the change in the potential caused by the change in strategy, then we have

$$\begin{aligned} \Delta\Phi &= \Phi(B_i, A_{-i}) - \Phi(A_i, A_{-i}) \\ &= \sum_{j\in\{k|B_{i,k}=1\wedge A_{i,k}=0\}} r_j(c_j(A)+1) - \sum_{j\in\{k|B_{i,k}=0\wedge A_{i,k}=1\}} r_j(c_j(A)) \end{aligned}$$

Thus we can conclude that $\Delta u_i = \Delta\Phi$. And thus the transaction inclusion game in We-TIPS is a potential game. The proof is thus completed. □

Since the transaction inclusion game in We-TIPS is a potential game, it always has a pure strategy Nash equilibrium and the finite improvement property [25], [26]. This implies that any asynchronous better response update process is guaranteed to reach a pure strategy Nash equilibrium. This motivates the algorithm design in the following section.

D. Transaction Inclusion Strategy in We-TIPS

The transaction inclusion strategy in We-TIPS is shown in Algorithm 2. The miner will analyze other miners' transaction selection strategy based on the weak block headers they broadcast, where $\sum_{i\neq i^*} W(i,j)$ in line 7 in Algorithm 2 denotes the number of miners who select the transaction j in their latest weak block header. After knowing other miners' transaction selection strategies, the miner will estimate the expected reward for each transaction based on Lemma 2 (line 8). Based on the estimation of each transaction, the miner will adopt a myopic strategy and select the transactions with the highest expected reward (line 1-5). In this way, the miner will adopt the best response to other miners' behaviors.

Algorithm 2: Transaction Inclusion Strategy in We-TIPS

Input: $i^*, \mathbf{f}, W, \lambda, \Delta$ // the miner index i^*; transaction fee $\mathbf{f}$; transaction selection matrix W; blockchain setting λ, Δ
Output: T // the set of selected transactions

1 **Function** `TransactionSelection`$(i^*, \boldsymbol{f}, W, \lambda, \Delta)$**:**
2 **for** $j = 1, \ldots, m$ **do**
3 Estimate the expected reward of transaction j, i.e., $e_j =$ `Estimate`$(i^*, \boldsymbol{f}, W, \lambda, \Delta)$
4 Select the transactions with the top-n reward as a set T
5 **Return** T

6 **Function** `Estimate`$(i^*, \boldsymbol{f}, W, \lambda, \Delta, j)$**:**
7 $c = \sum_{i\neq i^*} W(i,j) + 1$
8 $r = r_j(c)$ calculated by Lemma 2
9 **Return** r

We first show that Algorithm 2 can achieve the η-approximate Nash equilibrium in the following theorem.

Theorem 2. *Algorithm 2 can achieve the η-approximate Nash equilibrium, where*

$$\eta = O\left(\beta^{-1}N^2\log N\sum_{j=1}^{n} f_j\right).$$

Proof. We will show that Algorithm 2 will find a ϵ-approximate Nash equilibrium with the potential function:

$$\Phi(A) = \sum_{j=1}^{m}\sum_{k=1}^{c_j(A)} r_j(k) \leq \sum_{j=1}^{m} n_j(A) r_j(1) \leq N\sum_{j=1}^{n} f_j.$$

When the miners are homogeneous, each miner has the same probability $1/N$ to generate a new block. When a new weak block is mined, the corresponding miner can publish his latest transaction selection set. If the miner changes his transaction inclusion strategy, we decrease the potential function Φ by at least η. Otherwise, we have reached the η-approximate Nash equilibrium. Similar to the Coupon collector's problem [27], the expected number of rounds before each miner generates a weak block is $\theta(N\log N)$, where N is the number of miners. Therefore, before reaching the η-approximate equilibrium, each $\theta(N\log N)$ rounds will decrease Φ with a least η. Then the expected number of rounds is at most

$$\frac{\Phi(a)\theta(N\log N)}{\eta} \leq \epsilon^{-1}N^2\log N\sum_{j=1}^{n} f_j.$$

Besides, the game ends when a strong block is mined. With $\beta = T_w/T_s$ denoting the difficulty ratio of mining a strong block and weak block, the expected number of weak blocks in each round is β. Therefore, we have that

$$\theta\left(\eta^{-1}N^2\log N\sum_{j=1}^{n}f_j\right)\leq\theta(\beta).$$

Thus, we have

$$\eta=O\left(\beta^{-1}N^2\log N\sum_{j=1}^{n}f_j\right).$$

The proof is thus completed. □

According to Theorem 2, we can find that a larger weak block ratio β indicates a smaller η. Specifically, we can show that when β is large enough, i.e., $\beta\to\infty$, Algorithm 2 is guaranteed to achieve the pure strategy Nash equilibrium.

Theorem 3. *When the weak block ratio β is large enough, i.e., $\beta\to\infty$, Algorithm 2 is guaranteed to achieve the pure strategy Nash equilibrium with probability 1.*

Proof. Since the potential game has the finite improvement property, we assume that after χ better response update, it can reach the pure strategy Nash equilibrium. According to the Coupon collector's problem, the probability that there is no better response update in r rounds (each weak block corresponds to a single round) is that $\Pr(r)\leq\left(1-\frac{1}{N}\right)^r\leq e^{-r/N}$. And the probability that there will be χ better response update within $\chi\cdot r$ rounds is that $\Pr(\chi)\geq\left(1-e^{-r/N}\right)^{\chi}$. The game ends when a strong header is mined with the probability $1/\beta$. And the number of total rounds in the game follows the geometric distribution with the expected value $\mu=\beta$ and variance $\sigma^2=\beta^2-\beta$. Using Chebyshev's inequality, the probability that there will be at least $\chi\dot{r}$ rounds is

$$\Pr(\chi\cdot r)=1-\sum_{i=1}^{\chi\cdot r}(1-\frac{1}{\beta})^i\frac{1}{\beta}\geq\frac{\beta^2-\beta}{(\beta-\chi r)^2}.$$

Let $r=\frac{\sqrt{\beta}}{\chi}$, then the probability that Algorithm 2 will achieve the pure strategy Nash equilibrium is

$$\Pr(\text{NE})=\Pr(\chi\cdot r)\cdot\Pr(\chi)\geq\frac{\beta^2-\beta}{(\beta-\sqrt{\beta})^2}\cdot\left(1-e^{-\frac{\sqrt{\beta}}{N}}\right)^{\chi}.$$

Thus we have $\lim_{\beta\to\infty}\Pr(\text{NE})=1$, which implies that we can achieve the pure strategy Nash equilibrium with probability 1 when $\beta\to\infty$. The proof is thus completed. □

Consistent with the equilibrium analysis in TIPS, when the effective network propagation delay Δ is small enough, Algorithm 2 in We-TIPS is guaranteed to achieve the unique Nash equilibrium as in TIPS.

Theorem 4. *Selecting the transactions with the top-n transaction fee is the unique Nash equilibrium in this transaction inclusion game when $\Delta\leq\frac{1}{\lambda}\varphi\left(\frac{f_{n+1}}{f_n}\right)$ and Algorithm 2 is guaranteed to achieve the unique Nash equilibrium.*

Proof. Since $\Delta\leq\frac{1}{\lambda}\varphi\left(\frac{f_{n+1}}{f_n}\right)$ is equivalent to $\frac{1-e^{-\lambda\Delta}}{\lambda\Delta}\geq\frac{f_{n+1}}{f_n}$. We are going to prove that when $\frac{1-e^{-\lambda\Delta}}{\lambda\Delta}\geq\frac{f_{n+1}}{f_n}$, the unique Nash equilibrium in this transaction inclusion game is to always include the transactions with the highest transaction fee, i.e., the top n transactions. Initially, according to Algorithm 2, all the miners will choose to include the transactions with the highest transaction fee, and thus achieve the Nash equilibrium and will not change their transaction inclusion strategy.

Denote the transaction set $\mathbb{A}=\{1,2,\cdots,n\}$ as the top n transactions with the highest fee, and transaction set $\mathbb{B}=\{n+1,\cdots,m\}$ as the set of the remaining transactions. Then according to Lemma 2, the expected reward for the transaction in $\mathbb{A}$ is $r_j=r_j(N)=(1-e^{-\lambda\Delta})f_j/(\lambda\Delta),\forall j\in\mathbb{A}$, and the expected reward for transaction in $\mathbb{B}$ is $r_j=r_j(1)=f_j,\forall j\in\mathbb{B}$. Then we have that $\min_{j\in\mathbb{A}}r_j=r_n=\frac{1-e^{-\lambda\Delta}}{\lambda\Delta}f_n$. and that $\max_{j\in\mathbb{B}}r_j=r_{n+1}=f_{n+1}$. Since $\frac{1-e^{-\lambda\Delta}}{\lambda\Delta}\geq\frac{f_{n+1}}{f_n}$, we have that $\min_{j\in\mathbb{A}}\geq\max_{j\in\mathbb{B}}$, therefore transactions in $\mathbb{A}$ are always the transactions with the highest expected reward, none of the miners have the incentive to choose other transactions, which reaches the equilibrium. The proof is thus completed. □

V. Heterogeneous Miners

In this section, we delve into the dynamics of transaction inclusion strategies among miners with heterogeneous hash rates.

As an extension of Lemma 2, we now examine the anticipated reward for miner i to include transaction j in the context of a scenario involving heterogeneous miners.

Lemma 3. *The expected reward for miner i to include transaction j is*

$$r_j^i(\mathcal{C}_j^i,\boldsymbol{x})=\sum_{\mathcal{K}_i\in\mathcal{P}(\mathcal{N}_i)}\frac{(\lambda\Delta)^{|\mathcal{K}_i|}}{(|\mathcal{K}_i|)!}e^{-\lambda\Delta}\cdot\sum_{\pi\in\Pi(\mathcal{K}_i)}\prod_{k=1}^{|\mathcal{K}_i|}\frac{x_{\pi(k)}}{1-\sum_{j=1}^{k-1}x_{\pi(j)}}\cdot f_j/\left(|\mathcal{K}_i\cap\mathcal{C}_j^i|+1\right),$$

where $\mathcal{C}_j^i$ is the set of miners (except miner i itself) who decide to include transaction j in their newly-mined block, $\mathcal{N}_i=\mathcal{N}/\{i\}$ denote the set of miners except miner i. Let $\mathcal{P}(\mathcal{N}_i)$ denote as the power set of the set $\mathcal{N}_i$, which is the set of all the subsets of $\mathcal{N}_i$, including all the empty set and the set $\mathcal{N}_i$ itself. $\Pi(\mathcal{K}_i)$ denotes the set of all permutations of the elements of $\mathcal{K}_i$, and $\boldsymbol{x}=[x_1,\ldots,x_N]$ denotes the miners' normalized hash rate vector, i.e., $\sum_{i=1}^{N}x_i=1,x_i\in[0,1]$.

Proof. Let $\mathcal{N}_i=\mathcal{N}/\{i\}$ denote the set of miners except miner i. Let $\mathcal{P}(\mathcal{N}_i)$ denote as the power set of the set $\mathcal{N}_i$, which is the set of all the subsets of $\mathcal{N}_i$, including all the empty set and the set $\mathcal{N}_i$ itself.

We now investigate the expected reward for miner i to include transaction j, Denote the set $\mathcal{K}_i\subset\mathcal{N}_i$ as the set of miners (except miner i itself) who mine a new block during the block propagation time Δ. Since the block generation process follows the Poisson process with a rate λ, the probability that $|\mathcal{K}_i|$ blocks are generated during the block propagation time Δ is

$$\Pr(\#\text{block}=|\mathcal{K}_i|)=\frac{(\lambda\Delta)^{|\mathcal{K}_i|}}{(|\mathcal{K}_i|)!}e^{-\lambda\Delta}.$$

Then the problem of calculating the probability that the miners in $\mathcal{K}_i$ mine new blocks given the condition that $|\mathcal{K}_i|$ blocks will be generated is equivalent to the sampling problem with varying probability without replacement [24], where the normalized miners' hash rates $\mathbf{x}$ corresponds to their weights. Let $\Pi(\mathcal{K}_i)$ denote the set of all permutations of the elements of $\mathcal{K}_i$. Then the probability that the miners in $\mathcal{K}_i$ mine new blocks given the condition that $|\mathcal{K}_i|$ blocks will be generated is

$$\Pr(\mathcal{K}_i|\#\text{block} = |\mathcal{K}_i|) = \sum_{\pi\in\Pi(\mathcal{K}_i)} \prod_{k=1}^{|\mathcal{K}_i|} \frac{x_{\pi(k)}}{1-\sum_{j=1}^{k-1} x_{\pi(j)}}.$$

Denote $\mathcal{C}_j^i$ as the set of miners (except miner i itself) who decide to include transaction j in their newly-mined block and $\mathcal{C}_j^i \subset \mathcal{N}_i$. Therefore, at the final state s^*, there will be $|\mathcal{K}_i \cap \mathcal{C}_j^i| + 1$ miners who include transaction j in their newly-mined block. Then the expected reward for these miners on transaction j is $f_j/\left(|\mathcal{K}_i \cap \mathcal{C}_j^i| + 1\right)$. Therefore, the expected reward for miner i to include transaction j with $\mathcal{C}_j^i$ as the set of miners (except miner i itself) who decide to include transaction j in their newly-mined block is

$$\begin{aligned} r_j^i(\mathcal{C}_j^i, \mathbf{x}) &= \sum_{\mathcal{K}_i\in\mathcal{P}(\mathcal{N}_i)} \Pr(\#\text{block} = |\mathcal{K}_i|) \cdot \Pr(\mathcal{K}_i|\#\text{block} = |\mathcal{K}_i|) \\ &\quad \cdot f_j/\left(|\mathcal{K}_i \cap \mathcal{C}_i| + 1\right) \\ &= \sum_{\mathcal{K}_i\in\mathcal{P}(\mathcal{N}_i)} \frac{(\lambda\Delta)^{|\mathcal{K}_i|}}{(|\mathcal{K}_i|)!} e^{-\lambda\Delta} \cdot \sum_{\pi\in\Pi(\mathcal{K}_i)} \prod_{k=1}^{|\mathcal{K}_i|} \frac{x_{\pi(k)}}{1-\sum_{j=1}^{k-1} x_{\pi(j)}} \\ &\quad \cdot f_j/\left(|\mathcal{K}_i \cap \mathcal{C}_j^i| + 1\right). \end{aligned}$$

The proof is thus completed. □

The transaction inclusion strategy for heterogeneous miners in We-TIPS is delineated in Algorithm 3. To gauge the hash rates of other miners, each miner maintains a window that tracks the number of weak blocks mined by all participants. The ratio of weak blocks mined by a specific miner directly corresponds to its hash rate (line 8-9).

Leveraging these estimations of miners' hash rates, a miner can then project the expected reward for each transaction using Lemma 3 (line 10). Subsequently, the miner adopts a myopic strategy, strategically selecting transactions with the highest anticipated rewards (line 4). This approach enhances the miner's decision-making process, optimizing transaction selection based on real-time estimates of hash rates and maximizing potential rewards.

When miners exhibit heterogeneity, the transaction inclusion game no longer qualifies as a potential game. This deviation stems from the fact that the expected reward for a miner to include a specific transaction is not solely contingent upon the number of miners selecting the same transaction or the cumulative computing power dedicated to that transaction.

For instance, consider a scenario where a transaction is chosen by a miner with a 5% hash rate versus one chosen by a miner with a 50% hash rate. The expected reward for each transaction differs significantly, reflecting the varying impact of hash rates on block generation. Furthermore, the expected reward for a transaction selected by a miner with a 50% hash rate is not equivalent to that chosen by ten miners, each with a 5% hash rate. This distinction arises because the miner with a 50% hash rate can generate at most one block including the specific transaction, whereas the collective ten miners with 5% hash rates may produce multiple blocks, thereby diminishing the expected reward for that transaction.

Algorithm 3: Transaction inclusion strategy with heterogeneous miners in We-TIPS

Input: $i^*, \mathbf{f}, W, \lambda, \Delta$ // the miner index i^*; transaction fee $\mathbf{f}$; transaction selection matrix W; blockchain setting λ, Δ
Output: T // the set of selected transactions

1 **Function** `TransactionSelection`$(i^*, \boldsymbol{f}, W, \lambda, \Delta)$**:**
2 **for** $j = 1, \ldots, m$ **do**
3 Estimate the expected reward of transaction j, i.e., $e_j =$ `Estimate`$(i^*, \boldsymbol{f}, W, \lambda, \Delta)$
4 Select the transactions with the top-n reward as a set T
5 **Return** T

6 **Function** `Estimate`$(i^*, \boldsymbol{f}, W, \lambda, \Delta, j)$**:**
7 $\mathcal{C}_j^i = \{i \mid W(i,j) = 1\}/\{i\}$
8 $x_i \leftarrow \frac{\text{number of weak blocks mined by miner } i}{\text{total number of weak blocks}}$
9 $\mathbf{x} \leftarrow [x_1, \ldots, x_N]$
10 $r = r_j^i(\mathcal{C}_j^i, \mathbf{x})$ calculated by Lemma 3
11 **Return** r

Despite the transaction inclusion game losing its potential game status in heterogeneous settings, the dynamics resemble those of homogeneous scenarios. Consequently, we continue to employ the transaction inclusion algorithm wherein miners optimize their responses to other miners' actions. We will substantiate this approach through experiments detailed in Section VIII, demonstrating that Algorithm 3 remains highly effective even amidst miner heterogeneity.

VI. Sensitive Analysis on Weak Block Ratio

In this section, we conduct a sensitivity analysis on the weak block ratio β to investigate how variations in this parameter affect system performance.

We formalize our model by considering a system comprising N homogeneous miners, collectively denoted as the set $\mathcal{N}$. For analytical tractability, we assume the transaction pool contains sufficient transactions, with fee structures exhibiting approximate uniformity across the pool. Under these conditions, rational miners are incentivized to strategically select distinct transactions from the pool to mitigate transaction inclusion conflicts during block construction. This scenario models a setting similar to Ethereum, where the base fee constitutes the dominant portion of the transaction fee, while the priority fee is negligible. For example, in Ethereum [28], the current base fee is 93, and the highest priority fee is 3, resulting in a transaction fee range predominantly between [93, 96].

Furthermore, all miners adhere to Algorithm 2 in the transaction inclusion game. According to Algorithm 2, initially, all N miners select the first transaction bundle with the highest transaction fee. In each round, one miner is randomly chosen to generate a weak block header, revealing their transaction choice within this header. Subsequently, the other miners update their choices to the next available (i.e., not yet chosen and revealed) transaction bundle. The game concludes each

round with a probability of $1/\beta$, where $\beta \geq 1$. Thus, the stopping round is defined as $T \sim \mathrm{Geom}(1/\beta)$, where Geom denotes the geometric distribution. After the game concludes, if we randomly select $n \leq N$ miners to generate the strong blocks, the expected number of distinct transaction bundles among their choices is presented in the following lemma.

Lemma 4. *After the game stops, if we randomly select $n \leq N$ miners from $\mathcal{N}$, the expected number of distinct transaction bundles among their selections is*

$$d(n,\beta) = \sum_{k=1}^{N-1} P(K=k)\left[\frac{nk}{N} + 1 - \frac{\binom{k}{n}}{\binom{N}{n}}\right] + P(K=N)\, n,$$

where the distribution of K is given by

$$P(K=k) = \sum_{r=k}^{\infty} \frac{1}{\beta}\left(1-\frac{1}{\beta}\right)^{r-1} \frac{\binom{N}{k}\, k!\, S(r,k)}{N^r},$$

with $S(r,k)$ denoting the Stirling numbers of the second kind.

Proof. We divide the argument into several steps.

Step 1. Process Description. Initially, all N miners choose the first transaction bundle. In each round, a miner is selected uniformly at random (with replacement) from the set $\mathcal{N} = \{1, 2, \ldots, N\}$. If the miner is being revealed for the first time, its current transaction bundle becomes fixed and the remaining unrevealed miners update their selection to the next available transaction bundle. Thus, when exactly K distinct miners have been revealed:

- These K miners each hold a distinct transaction bundle (say, transaction bundles $1, 2, \ldots, K$);
- The remaining $N - K$ unrevealed miners all hold the same new transaction bundle (transaction bundle $K+1$).

Step 2. Stopping Time Distribution. Let T be the round in which the game stops. The stopping time T follows a geometric distribution with parameter $1/\beta$:

$$P(T=r) = \frac{1}{\beta}\left(1-\frac{1}{\beta}\right)^{r-1}, \quad r \geq 1.$$

Because a miner may be selected more than once, the number K of distinct revealed miners after r rounds is the number of distinct outcomes in r independent draws from $\{1, \ldots, N\}$.

Step 3. Distribution of K. Conditional on the game stopping in round $T = r$, each round involves an independent draw from the N miners, so the total number of possible sequences of selections over r rounds is N^r. To have exactly k distinct miners revealed in these r rounds, we proceed as follows:

1) **Choose the k miners:** There are $\binom{N}{k}$ ways to choose which k miners will be the ones that appear.
2) **Count valid sequences:** For the chosen k miners, we need to count the number of sequences of r rounds in which each of these miners appears at least once. This is equivalent to counting the number of surjective functions from a set of r rounds to a set of k miners. The number of such surjections is given by $k!\, S(r,k)$, where $S(r,k)$ denotes the Stirling numbers of the second kind.

Thus, the number of favorable sequences is

$$\binom{N}{k}\, k!\, S(r,k),$$

and hence the conditional probability is

$$P(K=k \mid T=r) = \frac{\binom{N}{k}\, k!\, S(r,k)}{N^r}, \quad \text{for } r \geq k.$$

Averaging over the geometric stopping time T (with $P(T=r) = \frac{1}{\beta}\left(1-\frac{1}{\beta}\right)^{r-1}$) yields the unconditional distribution

$$\begin{aligned} P(K=k) &= \sum_{r=k}^{\infty} P(T=r)\, P(K=k \mid T=r) \\ &= \sum_{r=k}^{\infty} \frac{1}{\beta}\left(1-\frac{1}{\beta}\right)^{r-1} \frac{\binom{N}{k}\, k!\, S(r,k)}{N^r}. \end{aligned}$$

Step 4. Counting Distinct Transaction Bundles in a Sample. After the game stops, if we randomly select n miners from the $\mathcal{N}$, let X be the number of miners in the sample that belong to the K revealed ones. Since these K miners are uniformly distributed among all N miners, X follows a hypergeometric distribution with mean

$$E[X \mid K] = \frac{nK}{N}.$$

The K revealed miners each hold a distinct transaction bundle, while the unrevealed miners (if any) all hold the same transaction bundle. Therefore, the number of distinct transaction bundles in the sample is

$$d(n \mid K) = \begin{cases} X+1, & \text{if } X < n, \\ n, & \text{if } X = n, \end{cases}$$

which can be succinctly written as

$$d(n \mid K) = X + \mathbf{1}_{\{X<n\}},$$

where $\mathbf{1}_{\{X<n\}}$ is the indicator function that equals 1 if $X < n$ (i.e., if at least one miner in the sample is unrevealed) and 0 otherwise. Hence, the conditional expectation is

$$E\big[d(n \mid K)\big] = E[X \mid K] + 1 - P(X=n \mid K) = \frac{nK}{N} + 1 - \frac{\binom{K}{n}}{\binom{N}{n}},$$

with the convention that $\frac{\binom{K}{n}}{\binom{N}{n}} = 0$ when $K < n$. Note that if $K = N$ (i.e., all miners are revealed), then $d(n \mid N) = n$.

Step 5. Averaging Over K. Averaging the conditional expectation over the distribution of K gives

$$d(n,\beta) = \sum_{k=1}^{N-1} P(K=k)\left[\frac{n\,k}{N} + 1 - \frac{\binom{k}{n}}{\binom{N}{n}}\right] + P(K=N)\, n\,.$$

This completes the proof. □

The monotonicity and limit of $d(n,\beta)$ is analyzed in the following lemma.

Lemma 5. *For fixed integers $1 \le n \le N$ and for $\beta \ge 1$, define*

$$d(n,\beta) = \sum_{k=1}^{N-1} P(K=k)\left[\frac{nk}{N} + 1 - \frac{\binom{k}{n}}{\binom{N}{n}}\right] + P(K=N)\, n,$$

with

$$P(K=k) = \sum_{r=k}^{\infty} \frac{1}{\beta}\Big(1-\frac{1}{\beta}\Big)^{r-1} \frac{\binom{N}{k}\, k!\, S(r,k)}{N^r},$$

where $S(r,k)$ are the Stirling numbers of the second kind. Then:

1) *The function $\beta \mapsto d(n,\beta)$ is monotone increasing.*
2) *In the limit as $\beta \to \infty$, we have $d(n,\beta) = n$.*

Proof. We begin by interpreting the model in the following steps.

Step 1. Representation via a Random Variable K. Let T denote the round in which the game stops. By design,

$$P(T=r) = \frac{1}{\beta}\Big(1-\frac{1}{\beta}\Big)^{r-1}, \quad r \ge 1,$$

so that T is geometrically distributed with mean β. In each round one miner is chosen uniformly at random from N miners (with replacement), and if the chosen miner is being revealed for the first time its choice is fixed. Thus, if $T = r$ then the number K of distinct miners that have been revealed satisfies

$$P(K=k \mid T=r) = \frac{\binom{N}{k}\, k!\, S(r,k)}{N^r}, \quad r \ge k,$$

so that the unconditional law is

$$P(K=k) = \sum_{r=k}^{\infty} \frac{1}{\beta}\Big(1-\frac{1}{\beta}\Big)^{r-1} \frac{\binom{N}{k}\, k!\, S(r,k)}{N^r}.$$

Notice that the mapping $r \mapsto K(r)$ is nondecreasing, so that larger values of T yield stochastically larger K. Moreover, observe that the bracket function

$$f(k) = \begin{cases} \dfrac{nk}{N} + 1 - \dfrac{\binom{k}{n}}{\binom{N}{n}}, & k < N, \\ n, & k = N, \end{cases}$$

which represents the number of distinct transaction bundles in a sample when $K = k$, is increasing in k (with the convention that for $k < n$ we take $\binom{k}{n} = 0$). Thus, we can write

$$d(n,\beta) = E[f(K)].$$

Step 2. Monotonicity in β. Consider two parameters $\beta_1 < \beta_2$. Then the corresponding geometric distributions have parameters

$$p_i = \frac{1}{\beta_i}, \quad q_i = 1 - \frac{1}{\beta_i}, \quad i = 1, 2.$$

Since $p_1 > p_2$ and $q_1 < q_2$, it is known that the geometric distribution with parameter p_2 is stochastically larger than that with parameter p_1; that is,

$$P_{\beta_1}(T \le t) \ge P_{\beta_2}(T \le t) \quad \text{for all } t \ge 1.$$

Since K is nondecreasing in T, it follows by standard arguments on the composition of stochastic orders that

$$P_{\beta_1}(K \le k) \ge P_{\beta_2}(K \le k) \quad \text{for all } k.$$

In other words, the distribution of K shifts to larger values as β increases. Since $f(k)$ is increasing in k, the expectation $E[f(K)]$ is larger under β_2 than under β_1. Therefore,

$$d(n,\beta) = E[f(K)]$$

is monotone increasing in β.

Step 3. The Limit as $\beta \to \infty$. As $\beta \to \infty$, the stopping probability per round $1/\beta$ tends to zero and the mean stopping time tends to infinity. Hence, with probability one the game continues until all N miners have been revealed. Formally, for every fixed $k < N$,

$$\lim_{\beta\to\infty} P(K=k) = 0,$$

while

$$\lim_{\beta\to\infty} P(K=N) = 1.$$

Therefore, taking the limit in the expression for $d(n,\beta)$ we obtain

$$\lim_{\beta\to\infty} d(n,\beta) = \lim_{\beta\to\infty} \left\{ \sum_{k=1}^{N-1} P(K=k) f(k) + P(K=N)\, n \right\} = n.$$

Conclusion. We have shown that $d(n,\beta)$ is an increasing function of β and that

$$\lim_{\beta\to\infty} d(n,\beta) = n.$$

This completes the proof. □

According to Lemma 4 and Lemma 5, we observe that $d(n,\beta)$ increases monotonically as β increases. As β approaches infinity, the stopping probability per round, $1/\beta$, converges to zero. In this scenario, $d(n,\beta)$ approaches n, which is the expected outcome, as each miner reveals and selects a distinct transaction bundle, thus eliminating any transaction inclusion collision. This observation aligns with the theoretical findings in Theorem 3, demonstrating that when the weak block ratio β is sufficiently large, specifically as $\beta \to \infty$, Algorithm 2 reliably attains the pure strategy Nash equilibrium with probability 1. Building on Lemma 4, the system utilization with N homogeneous miners and β weak block ratio is shown in the following theorem.

Theorem 5. *Consider a DAG-based blockchain system with We-TIPS protocol where blocks are generated according to a Poisson process with rate λ and the propagation delay is Δ, the weak block ratio is β, the expected system utilization is given by*

$$U(\beta) = \sum_{n=0}^{\infty} \frac{d(n+1,\beta)}{n+1} \frac{(\lambda\Delta)^n}{n!} e^{-\lambda\Delta},$$

where $d(n,\beta)$ is presented in Lemma 4.

Proof. Since the block generation process follows a Poisson process with rate λ, the probability that exactly n blocks are

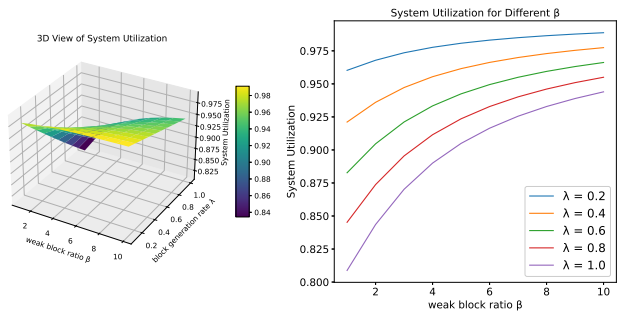


Fig. 2. System utilization with different weak block ratios.

generated during the propagation time Δ after one block is generated is

$$\Pr(\#\text{blocks} = n) = \frac{(\lambda\Delta)^n}{n!} e^{-\lambda\Delta}.$$

Given that $d(n, \beta)$ is the expected number of distinct transaction bundles among the n blocks, the utilization when n extra blocks are generated during the propagation time Δ is

$$U(n, \beta) = \frac{d(n+1, \beta)}{n+1}.$$

To obtain the overall expected utilization, we average $U(n)$ over the distribution of n blocks generated during Δ:

$$\begin{aligned} U(\beta) &= \sum_{n=1}^{\infty} U(n, \beta) \Pr(\#\text{blocks} = n) \\ &= \sum_{n=1}^{\infty} \frac{d(n+1, \beta)}{n+1} \frac{(\lambda\Delta)^n}{n!} e^{-\lambda\Delta}. \end{aligned}$$

This completes the proof. □

According to Theorem 5 and Lemma 5, the system utilization $U(\beta)$ increases monotonically with the weak block ratio β, implying that a larger weak block ratio can help achieve the Nash equilibrium, preventing transaction inclusion conflicts, and enhancing system performance. Moreover, as the weak block ratio β becomes large, specifically as $\beta \to \infty$, system utilization approaches 100%, i.e., $\lim_{\beta\to\infty} U(\beta) = 1$, which aligns with Theorem 3.

Figure 2 illustrates the relationship between system utilization, weak block ratio (β), and block generation rate (λ). The left panel presents a 3D view, depicting how system utilization varies with changes in β and λ. Utilization levels are color-coded from purple (lower utilization) to yellow (higher utilization). The right panel displays 2D plots of system utilization as a function of β for various fixed values of λ: 0.2, 0.4, 0.6, 0.8, and 1.0, with distinct colors for clarity. As β increases, system utilization consistently rises across all λ values, emphasizing the impact of the weak block ratio on system performance.

VII. Security Analysis

A. Incentive Compatibility of Honest Weak Header Broadcasting

In We-TIPS, each miner broadcasts a weak header that includes a Bloom filter indicating its intended transaction set. This signal enables other miners to adjust their transaction inclusion strategies to avoid collision. In this subsection, we prove that if a miner deviates from honest broadcasting, either by withholding (no-broadcast) or by misreporting its transaction set (misleading broadcast), its expected revenue will be reduced. Hence, when all other miners broadcast honestly, honest weak header broadcasting is a Nash equilibrium.

Theorem 6. *If a miner deviates from honest broadcasting (either by withholding its weak header or by broadcasting a misleading weak header), then its expected reward will be lower. Therefore, honest broadcasting is a Nash equilibrium.*

Proof. Under the honest strategy, miner i broadcasts its true intended transaction set T_i in its weak header. In response, the other miners adjust their transaction inclusion strategies to avoid transactions in T_i, thereby reducing the chance of collision. Consequently, if miner i mines a strong block, each transaction $j \in T_i$ is included by exactly c_j miners (with miner i among them), and its share of the fee is given by $r_j = \frac{f_j}{c_j}$. Let the total revenue of miner i be the sum over all transactions in T_i.

Now consider the case where miner i deviates from the honest strategy. We distinguish between two types of deviation:

1) **No-broadcasting:** Miner i withholds its weak header entirely. In this case, other miners receive no signal regarding miner i's intended transaction set. As a result, these miners may include some or all transactions from T_i in their own blocks, increasing the overall competition for each such transaction.
2) **Misleading broadcast:** Miner i broadcasts a weak header that reports a transaction set $\bar{T}_i \neq T_i$. (Note that the case $\bar{T}_i = \emptyset$ is equivalent to no-broadcasting.) Here, although miner i is actually mining on T_i, it publicly signals only $\bar{T}_i$. Thus, other miners adjust their strategies based on the reported set $\bar{T}_i$ rather than the true set T_i.

We denote by r_j the estimated value of transaction j in Algorithm 2 when miner i is honest and by $\bar{r}_j$ the estimated value when miner i broadcasts $\bar{T}_i$. The effect of deviation can be analyzed by partitioning T_i into two disjoint subsets:

- For transactions in $T_i \cap \bar{T}_i$: Since miner i's broadcast agrees with its true intention on these transactions, the estimation in Algorithm 2 remains unchanged (i.e., $\bar{r}_j = r_j$). Consequently, the number of miners including such a transaction remains the same, and miner i's expected reward on these transactions is unaffected.
- For transactions in $T_i \setminus \bar{T}_i$: Since these transactions are not signaled in the weak header, the count of miners' reported intentions (as computed in Line 7 of Algorithm 2) is lower. In turn, this leads to a higher estimated value on Line 8, making these transactions appear more attractive to other miners. Therefore, more miners are likely to include any $j \in T_i \setminus \bar{T}_i$, which increases the effective inclusion count for these transactions. As a result, miner i's share of the fee for such transactions decreases (i.e., its reward becomes $f_j/\bar{c}_j$ with $\bar{c}_j > c_j$).

Recall that the reward for mining is obtained only from winning a strong block (since generating a weak header does

not yield direct rewards), and the revenue from each transaction is inversely related to the number of miners including that transaction. Therefore, any deviation that causes even a subset of T_i to be under-reported (or not reported) increases the competition on those transactions, thereby reducing miner i's expected revenue.

Summing over all transactions in T_i, the overall expected revenue under deviation is strictly lower than that under honest broadcasting. Consequently, given that all other miners are honest, miner i has no incentive to deviate. This completes the proof that honest weak header broadcasting is a Nash equilibrium in the weak header signaling subgame. □

Theorem 6 establishes that honestly broadcasting weak block headers constitutes a Nash equilibrium in the transaction inclusion game with We-TIPS. This is primarily because weak block headers serve solely as signals during the mining process and do not influence the actual mining reward. The mining reward is determined exclusively by successfully mined strong blocks. Broadcasting weak block headers honestly helps miners prevent others from selecting the same transactions, whereas withholding or broadcasting misleading weak headers does not offer this advantage.

Moreover, the effort required to generate any weak block header involves a significant computational cost due to the necessity of solving a partial proof-of-work (PoW) puzzle. This computational demand naturally deters the creation of malicious weak block headers, as the resources required would exceed any potential disruptive benefit.

B. Weak Block Header Spamming Attack

In this subsection, we address the potential concern of a weak block header spamming attack within our system, which we argue is largely impractical due to several key factors.

Firstly, the generation of weak block headers inherently requires computational effort, as each weak header contains a partial proof of work. This mechanism ensures that weak headers cannot be effortlessly produced by any malicious entities. The proof of work design intrinsic to our protocol acts as a deterrent against prolific header generation, as it requires dedicated computational resources.

Secondly, the frequency and number of weak block headers are controlled by the system parameter known as the weak block ratio, β. In practice, this parameter will not be set too high, as an unreasonably large weak block ratio would be impractical and might lead to network inefficiencies. Our empirical findings suggest that moderate values of β are sufficient to achieve optimal coordination without risking network congestion or an increase in computational demand.

Moreover, as noted by [12], the size of a block header is less than 1% of the original block size, comparable to the size of a typical transaction. Hence, the additional bandwidth consumed by broadcasting weak headers is negligible in the context of the total network traffic. This minimal bandwidth overhead implies that even under conditions where weak headers are broadcasted more frequently, the impact on overall network performance remains insignificant.

In summary, the potential for a weak block header spamming attack is limited by the computational constraints required to generate weak headers, the practical configuration of the weak block ratio β, and the negligible bandwidth impact of header broadcast. These factors collectively ensure that such an attack is not a viable concern within the framework of our protocol.

C. Switching Weak Header Mining

A miner can first mine on the transactions with the highest transaction fees. Once the miner discovers a weak header, they can switch to mining other transactions, assuming that other miners will not mine the transaction already included in a weak header. This approach potentially leads to higher revenue for the miner, but it may result in delays for transactions with high transaction fees. This behavior bears resemblance to the delay-of-service attack discussed in TIPS [12]. It is important to note that this strategic behavior does not impact the system throughput or the consensus mechanism, though further investigation could be interesting.

VIII. Performance Evaluation

In this section, we first provide empirical results of Conflux [15], one of the most popular DAG-based blockchain systems, and show how transaction inclusion collision degrades its system performance. We conduct experiments to demonstrate the performance of We-TIPS, and further validate our analysis.

A. Empirical Results of Conflux

Conflux is one of the most popular DAG-based blockchain systems. To avoid transaction inclusion collision, Conflux adopts the random transaction inclusion strategy with transaction fee priority, that is, the miner will include the transaction i in his block with probability p_i, where $\frac{p_1}{f_1} = \frac{p_2}{f_2} = \cdots = \frac{p_m}{f_m}$.

We set up a full node of Conflux on a server with one AMD Ryzen 5950X (16 cores 32 HT) CPU and 32 GB memory to pull the latest blocks from the mainnet of Conflux. We have collected the blocks in 1000 epochs (from 32289102-th epoch to 32290102-th epoch), which includes total of 5584 transactions but only 4043 unique transactions, implying that the block capacity utilization of Conflux is around 72.40%. This implies that around 27.60% block capacity is wasted due to the transaction inclusion collision.

B. System Implementation

We developed a high-fidelity DAG-based blockchain simulator using Python and the SimPy framework [29] based on the implementation of the Phantom (GhostDAG) block-DAG protocol [30]. Our simulator implements a comprehensive protocol stack that includes both standard and advanced transaction inclusion strategies, making it a versatile tool for analyzing the performance and security of DAG-based blockchain systems.

The simulator faithfully incorporates the concepts of the block propagation model, transaction inclusion strategies, and the use of weak blocks to enhance signaling during the

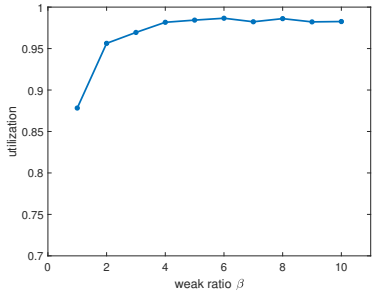


Fig. 3. Utilization of We-TIPS with different ratios β

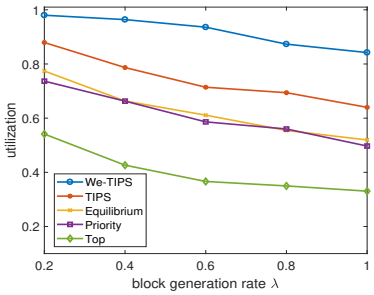


Fig. 4. Utilization of different transaction inclusion protocols

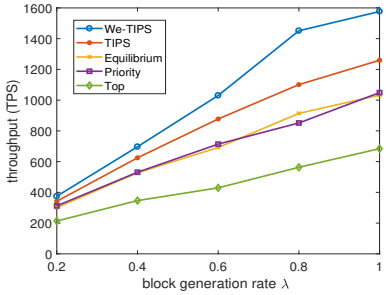


Fig. 5. TPS of different transaction inclusion protocols

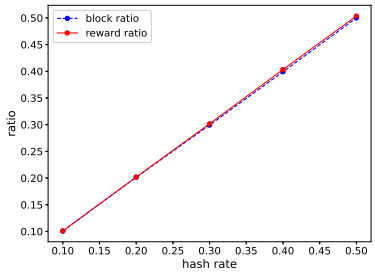


Fig. 6. Reward ratio with different hash rates

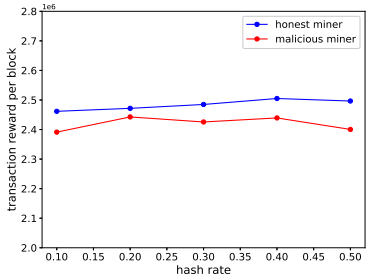


Fig. 7. Transaction reward per block with different hash rates

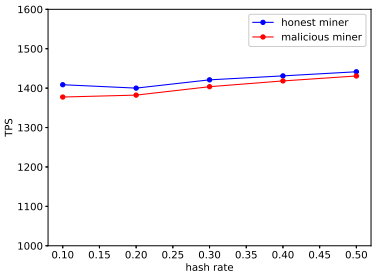


Fig. 8. TPS with different hash rates

mining process. We utilize the "pybloom-live" package [31] to efficiently implement Bloom filter data structures, which play a critical role in transaction selection and block propagation.

In our implementation, the blockchain network comprises multiple miners, each maintaining its own local transaction pool. These miners are fully connected, forming a peer-to-peer network where each node operates independently based on its local information. This setup models the real-world scenarios of decentralized networks, where block creation and propagation are influenced by network latency and miners' strategic behaviors.

Each miner continuously monitors the network for weak block signals, which provide partial proof-of-work and indicate current transaction selections. Upon receiving a weak block header, miners update their transaction pools, modifying their strategies to mitigate transaction inclusion collisions, thereby optimizing resource use and improving system throughput.

Our simulator supports diverse transaction inclusion strategies, adapting techniques from conventional linear blockchains to the unique demands of DAG structures. It simulates heterogeneous conditions where miners have varying hash rates, allowing comprehensive analysis across different network dynamics and configurations.

We demonstrate the robustness of our simulator by comparing traditional approaches like those of Conflux [15] and advanced protocols like TIPS [12]. These experiments validated the improved performance and efficiency of the weak block signaling approach under various conditions, emphasizing the importance of strategic transaction selection in achieving high block utilization and transaction throughput.

C. Experiment Configuration

We develop a DAG-based blockchain simulator in Python using Simpy [29]. The experimental configuration is as follows. We set the block size to 1MB, which is the current block size limitation in Bitcoin. With the average transaction size being 500 bytes, we put 2000 transactions in one block, i.e., $n = 2000$. Besides, we assume the size of the transaction pool to be $m = 10000$. The propagation delay for the whole block is a random variable following the normal distribution with the expectation of $\Delta = 10$, and the propagation delay for the signal is a random variable following the normal distribution with the expectation of $\tau = 1$. The block generation rate of the DAG-based blockchain system λ ranges from 0.1 to 1.

D. Weak Block Ratio Design

Since the weak block ratio β will directly affect the performance of We-TIPS as shown in Theorem 2, we first investigate the impact of the weak block ratio β. Figure 3 demonstrates the block utilization of We-TIPS with different weak block ratios β. Specifically, when $\beta = 1$, the We-TIPS degenerates to TIPS. We can find that block utilization increases with the weak block ratio. This is because a large β can provide more accurate information for miners to avoid transaction inclusion collision. However, due to the propagation delay of the weak block header, when β is large, too many weak block headers may be generated during the mining process, resulting in an intensive network load. And some weak block headers may not be broadcast to all the miners in time, leading to insignificant improvement. As shown in Figure 3, $\beta = 4$ is the "elbow" where the insignificant increase in the utilization is no longer

worth the additional increase on β. Thus, we take $\beta = 4$ in our experiment.

E. Performance Results

To compare and further demonstrate the performance of We-TIPS, we consider the following baselines in DAG-based blockchain systems:

- "TIPS": stands for transaction inclusion protocol in [12].
- "Priority": stands for the random strategy with transaction fee priority adopted by Conflux [15].
- "Equilibrium": stands for the equilibrium strategy in the standard DAG-based blockchain system [10].
- "Top": stands for the top-n strategy in the standard DAG-based blockchain system, where the miners always include the transactions with the highest transaction fees.
- "We-TIPS": stands for our proposed weak-block-based transaction inclusion protocol with signaling.

Figure 4 shows the block capacity utilization of different transaction inclusion protocols. We can find that We-TIPS always achieves the highest utilization compared to TIPS and the other protocols. Specifically, when $\lambda = 0.2$, the "Priority" strategy can achieve 77% utilization, the "Equilibrium" strategy can achieve 79% utilization, TIPS can achieve 90% utilization, while We-TIPS can achieve the astonishing 98% utilization. This implies that We-TIPS can effectively avoid transaction inclusion collision. Figure 5 shows the system throughput (TPS) of different transaction inclusion protocols, which shows that We-TIPS also always achieves the highest system throughput compared to TIPS and the other protocols.

F. Fairness on Mining Reward

To further explore the fairness of mining rewards among heterogeneous miners under the We-TIPS protocol, particularly focusing on the transaction inclusion strategy outlined in Section V, we conducted experiments involving five different miners. Among these, one miner has a substantial hash rate of 50%, while the remaining 10 miners exhibit homogeneity with a hash rate of 5% each.

fee range	hash rate	block ratio	transaction ratio	reward ratio
[10,110]	50%	0.49944995	0.499499494	0.515830256
	5%	0.05005501	0.050050051	0.048416974
[93,96]	50%	0.50087742	0.500589948	0.501256357
	5%	0.04991226	0.049941005	0.049874364

TABLE I
MINING REWARD RATIO WITH DIFFERENT TRANSACTION FEE RANGES

In Table I, we illustrate the mining reward ratios for all miners across different transaction fee ranges. The results indicate that the number of blocks and transactions mined by each participant correlates with their respective hash rates. Notably, miners with larger hash rates tend to earn slightly higher rewards, creating a disadvantage for those with smaller hash rates. This discrepancy arises because miners with greater hash rates enjoy an advantage in the transaction inclusion game, enabling them to prioritize higher fee transactions. Conversely, miners with lower hash rates face challenges in competing for transactions with higher fees, opting to avoid collisions and selecting transactions with lower fees.

Fortunately, this reward skew tends to diminish when the transaction fee range is more limited. Consider Ethereum's transaction fee structure after the adoption of the EIP-1559 mechanism, which divides fees into a base fee and a priority fee. This design ensures that the transaction fee range remains constrained. For instance, in Ethereum, currently, the base fee is 93, and the highest priority fee is 3 [28], resulting in a transaction fee range predominantly between [93, 96]. Table I depicts the reward ratio under this refined transaction fee setting, demonstrating the elimination of skew in mining rewards.

We further examine a scenario characterized by dynamic changes in hash rate, where the hash rate of one miner increases from 10% to 50%. We then analyze the reward ratio of this miner under various hash rates. As illustrated in Figure 6, the block ratio corresponds to the miner's hash rate. We observe that as the hash rate increases, the miner gains a relative advantage in transaction inclusion, which results in a reward ratio that is slightly higher than the block ratio. This discrepancy grows as the miner's hash rate increases. However, these advantages in transaction inclusion are generally minor, indicating that the We-TIPS protocol can sustain a degree of reward fairness throughout the mining process. This fairness in rewards also contributes to encouraging decentralization.

G. Incentive Evaluation and Impact of Malicious Behavior

We delve into a scenario where miners operate with differing hash rates: one miner employs a unique hash rate while the other miners collectively share the remaining hash rate. Our primary objective is to examine the contrasting behaviors of honest miners, those who comply with the We-TIPS protocol, and malicious miners who opt not to broadcast their weak block headers. This analysis provides valuable insights into the incentives for miners to comply with the We-TIPS protocol, as well as the ramifications of adversarial actions on the overall performance of the system.

Figure 7 illustrates the transaction reward per block under these conditions. It is evident that honest broadcasting of weak headers consistently yields higher rewards for miners. This outcome underscores the incentive compatibility of the We-TIPS protocol, affirming that adherence to honest broadcasting strategies is beneficial. Conversely, non-broadcasting by malicious miners leads to a decline in their performance.

Additionally, our findings indicate that the transaction reward per block remains stable across varying hash rates, further demonstrating that We-TIPS can effectively maintain fairness even when hash rates differ among miners.

Turning to system throughput, Figure 8 presents the transactions per second (TPS). Although a malicious miner might deploy a non-broadcast strategy intending to undermine system performance, the impact on TPS is marginal. The slight decrease in TPS arises because malicious miners do not disclose their currently selected transactions, increasing the likelihood of transaction inclusion collisions with honest miners. However, as depicted in Figure 8, this effect on TPS is negligible,

highlighting the robustness of the We-TIPS protocol against such adversarial tactics.

These insights collectively affirm that the We-TIPS protocol not only incentivizes honest behavior but also maintains system integrity and performance in the presence of varying miner behaviors and potential adversarial actions.

IX. Discussion

In this section, we discuss several practical aspects of implementing the We-TIPS protocol, including its computational and network overhead as well as its impact on security. We also provide additional perspectives on how the protocol enhances overall system performance.

A. Computational and Network Overhead

A primary concern when introducing any new protocol is the additional cost it may impose on the system. In the case of We-TIPS, the generation of weak block headers incurs no extra computational cost. These weak headers are produced naturally as a byproduct of the standard mining process. Much like in Bitcoin, where miners intermittently produce intermediate "weak blocks" or mining shares while searching for a valid (strong) block [32], [33], We-TIPS leverages these already occurring weak block headers to broadcast transaction inclusion signals. This design choice ensures that miners are not required to perform any extra work beyond their regular mining operations.

In addition, while the use of weak block headers introduces additional messages into the network, the associated network overhead is negligible. The size of a block header is minimal compared to that of a full block. According to [12], the size of the block header is less than 1% of the original block size. So the extra bandwidth consumed by broadcasting weak headers is insignificant relative to the overall network traffic. Thus, both the computational and network costs of integrating We-TIPS remain minimal.

B. Security Considerations

Security is of paramount importance in blockchain systems, and any protocol modification must be scrutinized for potential vulnerabilities. A natural concern is whether the use of weak block headers could introduce new attack vectors. However, a key design feature of We-TIPS is that weak block headers are used exclusively for signaling transaction inclusion and play no role in the consensus process. Consensus updates occur solely through strong blocks, i.e., headers that meet the full proof-of-work target T_s.

This isolation ensures that even if adversarial nodes attempt to broadcast misleading or incorrect weak block headers, the integrity of the consensus mechanism remains uncompromised. The security of the system against attacks such as the 51% attack is preserved because it is entirely determined by the difficulty target T_s, which is unchanged by the introduction of signaling. In effect, We-TIPS maintains the same robust security guarantees as traditional proof-of-work systems while simultaneously enhancing transaction inclusion efficiency.

C. Implications for Performance and Scalability

Beyond its negligible overhead and robust security, We-TIPS offers significant improvements in overall system performance. By enabling near real-time signaling through weak block headers, the protocol effectively mitigates transaction inclusion collisions. This coordination leads to better block capacity utilization and increased throughput, as miners can dynamically adjust their transaction selection strategies based on timely network information. In high-concurrency environments, such as those found in DAG-based blockchains, this improvement can result in a substantial enhancement of system efficiency.

Moreover, the modular design of We-TIPS allows for seamless integration with existing DAG-based blockchain systems. Since the protocol acts as an add-on component that does not interfere with the core consensus mechanism, current implementations can adopt We-TIPS with minimal changes. This compatibility not only facilitates easier deployment but also contributes to the scalability of the overall network by reducing redundant transaction processing.

X. Related Work

A. Transaction Inclusion Collision in DAG-based Blockchain

In the inclusive blockchain protocols [10], the authors model the transaction selection as a non-cooperative game with incomplete information and propose a myopic strategy. Conflux [15] models the transaction selection as a cooperative game and distributes the transaction fee based on Shapley value [34]. In [12], the authors propose "TIPS", which introduces a "signal" in the transaction selection game, and help to improve the system performance by avoiding transaction inclusion collision after finding a new block. However, TIPS only signals other miners when a new block is successfully mined, which limits its performance improvement. Faced with this limitation, we propose "We-TIPS", the weak-block-based transaction inclusion protocol with signaling in DAG-based blockchain, which can signal the miners to avoid transaction inclusion collision even during the mining process.

B. Weak Block

Employing weak solutions (and their variations) in Bitcoin is an idea circulating on Bitcoin forums for many years [32], [33]. Initial proposals leverage weak solutions (i.e., weak blocks) for faster transaction confirmations [35], [36] and fork selection rule [37]. Rizun proposes Subchains [38], where a chain of weak blocks bridging each pair of subsequent strong blocks is created. In [22], the authors propose the StrongChain based on the idea of the weak block to make the mining process more transparent and collaborative. However, the previous work mainly focuses on the application of weak blocks in the linear blockchain. Along a different line, we apply the weak block in DAG-based blockchain system and design a novel signaling protocol.

XI. Conclusion

In this paper, we proposed a novel weak-block-based transaction inclusion protocol with signaling, We-TIPS, which effectively avoids the transaction inclusion collision and achieves near-optimal performance while maintaining system security. Both the theoretical analysis and experiment results significantly demonstrate the high efficiency of We-TIPS.

Multi-agent reinforcement learning (MARL) is an ideal tool to capture the feature of the mining process in We-TIPS after modeling the system as a multi-agent Markov decision process (MMDP). Using MARL to explore a more efficient transaction inclusion strategy will be one of our future works.

References

[1] C. Chen and Z. Fang, "We-tips: Weak-block-based transaction inclusion protocol with signaling in dag-based blockchain," in *2023 21st International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. IEEE, 2023, pp. 191–198.

[2] L. Chettri and R. Bera, "A comprehensive survey on internet of things (iot) toward 5g wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, 2019.

[3] L. Duan, L. Huang, C. Langbort, A. Pozdnukhov, J. Walrand, and L. Zhang, "Human-in-the-loop mobile networks: A survey of recent advancements," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 4, pp. 813–831, 2017.

[4] L. Da Xu, Y. Lu, and L. Li, "Embedding blockchain technology into iot for security: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 452–10 473, 2021.

[5] M. Swan, J. Potts, S. Takagi, F. Witte, and P. Tasca, *Blockchain economics: implications of distributed ledgers: markets, communications networks, and algorithmic reality*. World Scientific Publishing Co. Pte. Ltd., 2019.

[6] T. Wang, Q. Wang, Z. Shen, Z. Jia, and Z. Shao, "Understanding characteristics and system implications of DAG-based blockchain in IoT environments," *IEEE Internet of Things Journal*, 2021.

[7] B. Cao, Y. Li, L. Zhang, L. Zhang, S. Mumtaz, Z. Zhou, and M. Peng, "When internet of things meets blockchain: Challenges in distributed consensus," *IEEE Network*, vol. 33, no. 6, pp. 133–139, 2019.

[8] H. Zhang, S. Leng, F. Wu, and H. Chai, "A DAG blockchain enhanced user-autonomy spectrum sharing framework for 6G-enabled IoT," *IEEE Internet of Things Journal*, 2021.

[9] Q. Wang, J. Yu, S. Chen, and Y. Xiang, "Sok: Diving into dag-based blockchain systems," *arXiv preprint arXiv:2012.06128*, 2020.

[10] Y. Lewenberg, Y. Sompolinsky, and A. Zohar, "Inclusive block chain protocols," in *International Conference on Financial Cryptography and Data Security*. Springer, 2015, pp. 528–547.

[11] H. Gupta and D. Janakiram, "CDAG: A serialized blockdag for permissioned blockchain," *arXiv preprint arXiv:1910.08547*, 2019.

[12] C. Chen, X. Chen, and Z. Fang, "Tips: Transaction inclusion protocol with signaling in dag-based blockchain," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 12, pp. 3685–3701, 2022.

[13] B. Fisch, R. Pass, and A. Shelat, "Socially optimal mining pools," in *Web and Internet Economics: 13th International Conference, WINE 2017, Bangalore, India, December 17–20, 2017, Proceedings 13*. Springer, 2017, pp. 205–218.

[14] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Satoshi Nakamoto*, 2008.

[15] C. Li, P. Li, D. Zhou, Z. Yang, M. Wu, G. Yang, W. Xu, F. Long, and A. C.-C. Yao, "A decentralized blockchain with high throughput and fast confirmation," in *2020 {USENIX} Annual Technical Conference ({USENIX}{ATC} 20)*, 2020, pp. 515–528.

[16] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, "Spectre: A fast and scalable cryptocurrency protocol," *Cryptology ePrint Archive*, 2016.

[17] Y. Sompolinsky and A. Zohar, "Phantom," *IACR Cryptology ePrint Archive, Report 2018/104*, 2018.

[18] A. Churyumov, "Byteball: A decentralized system for storage and transfer of value," *URL https://byteball. org/Byteball. pdf*, p. 11, 2016.

[19] C. LeMahieu, "Nano: A feeless distributed cryptocurrency network," *Nano [Online resource]. URL: https://nano. org/en/whitepaper (date of access: 24.03. 2018)*, vol. 16, p. 17, 2018.

[20] T. Rocket, M. Yin, K. Sekniqi, R. van Renesse, and E. G. Sirer, "Scalable and probabilistic leaderless bft consensus through metastability," *arXiv preprint arXiv:1906.08936*, 2019.

[21] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[22] P. Szalachowski, D. Reijsbergen, I. Homoliak, and S. Sun, "Strongchain: Transparent and collaborative proof-of-work consensus," in *28th {USENIX} Security Symposium ({USENIX} Security 19)*, 2019, pp. 819–836.

[23] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *International conference on financial cryptography and data security*. Springer, 2014, pp. 436–454.

[24] B. Rosen, "Asymptotic theory for successive sampling with varying probabilities without replacement, i," *The Annals of Mathematical Statistics*, pp. 373–397, 1972.

[25] J. N. Webb, *Game theory: decisions, interaction and Evolution*. Springer Science & Business Media, 2007.

[26] Y. Zhang and M. Guizani, *Game theory for wireless communications and networking*. CRC press, 2011.

[27] I. Adler, S. Oren, and S. M. Ross, "The coupon-collector's problem revisited," *Journal of Applied Probability*, vol. 40, no. 2, pp. 513–518, 2003.

[28] Ethereum gas tracker. [Online]. Available: https://etherscan.io/gastracker

[29] N. Matloff, "Introduction to discrete-event simulation and the simpy language," *Davis, CA. Dept of Computer Science. University of California at Davis. Retrieved on August*, vol. 2, no. 2009, pp. 1–33, 2008.

[30] A. Yaish. Phantom (ghostdag). [Online]. Available: https://github.com/AvivYaish/PHANTOM

[31] pybloom-live 3.1.0. [Online]. Available: https://pypi.org/project/pybloom-live/

[32] K. Rosenbaum. (2016) Weak blocks – the good and the bad. [Online]. Available: https://popeller.io/index.php/2016/01/19/weak-blocks-the-good-and-the-bad/

[33] G. Andresen. (2015) [bitcoin-dev] weak block thoughts. [Online]. Available: https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2015-September/011157.html

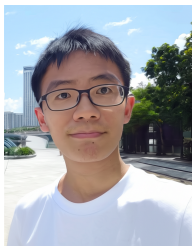
[34] A. E. Roth, *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press, 1988.

[35] T. Nolan. (2013) Decoupling transactions and POW. [Online]. Available: https://bitcointalk.org/index.php?topic=179598.0

[36] P. Todd. (2013) [bitcoin-development] near-block broadcasts for proof of tx propagation. [Online]. Available: https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2013-September/003275.html

[37] T. Nolan. (2013) [bitcoin-development] distributing low pow headers. [Online]. Available: https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2013-July/002976.html

[38] P. R. Rizun, "Subchains: A technique to scale Bitcoin and improve the user experience," *Ledger*, vol. 1, pp. 38–52, 2016.



Canhui Chen received the B.Eng. degree in software engineering from Sun Yat-sen University, Guangzhou, China, in 2021. Now he is currently pursuing a PhD degree in computer science at the Institute for Interdisciplinary Information Sciences (IIIS) of Tsinghua University. His current research interests include blockchain system and game theory.

Zhixuan Fang is a tenure-track assistant professor at the Institute for Interdisciplinary Information Sciences (IIIS) at Tsinghua University, Beijing, China. He mainly focuses on the design and analysis of multi-agent systems and networked systems. He received his Ph.D. degree in computer science from Tsinghua University, China, in 2018, and his B.S. degree in physics from Peking University, China, in 2013.